Linux Permissions Cheat Sheet

I created this repository in hopes that it may be used as a helpful reference.

Permissions

Permissions on Unix and other systems like it are split into three classes:

- User
- Group
- Other

Files and directories are owned by a **user**.

Files and directories are also assigned to a **group**.

If a user is not the owner, nor a member of the group, then they are classified as **other**.

Changing permissions

In order to change permissions, we need to first understand the two notations of permissions.

1. Symbolic notation
2. Octal notation

Symbolic notation

Symbolic notation is what you'd see on the left-hand side if you ran a command like ls -l in a terminal.
The first character in symbolic notation indicates the *file type* and isn't related to permissions in any way. The remaining characters are in sets of three, each representing a class of permissions.

The first class is the **user** class. The second class is the **group** class. The third class is the **other** class.

Each of the three characters for a class represents the read, write and execute permissions.

- r will be displayed if reading is permitted
- w will be displayed if writing is permitted
- x will be displayed if execution is permitted
- \- will be displayed in the place of r, w, and x, if the respective permission is *not* permitted

Here are some examples of symbolic notation:

- -rwxr--r--: A regular file whose **user** class has read/write/execute, **group** class has only read permissions, **other** class has only read permissions
- drw-rw-r--: A directory whose **user** class has read/write permissions, **group** class has read/write permissions, **other** class has only read permissions
- crwxrw-r--: A character special file whose **user** has read/write/execute permissions, **group** class has read/write permissions, **other** class has only read permissions

Octal notation

Octal (base-8) notation consists of at least 3 digits (sometimes 4, the left-most digit, which represents the setuid bit, the setgid bit, and the sticky bit).

Each of the three right-most digits are the sum of its component bits in the binary numeral system.

For example:

- The read bit (r in symbolic notation) adds 4 to its total
- The write bit (w in symbolic notation) adds 2 to its total
- The execute bit (x in symbolic notation) adds 1 to its total

So what number would you use if you wanted to set a permission to read and write? 4 + 2 = 6.

Symbolic notation	Octal notation	Plain English
-rwxr--r--	0744	**user** class can read/write/execute; **group** class can read; **other** class can read
-rw-rw-r--	0664	**user** class can read/write; **group** class can read/write; **other** class can read
-rwxrwxr--	0774	**user** class can read/write/execute; **group** class can read/write/execute; **other** class can read
----------	0000	None of the classes have permissions
-rwx------	0700	**user** class can read/write/execute; **group** class has no permissions; **other** class has no permissions
-rwxrwxrwx	0777	All classes can read/write/execute
-rw-rw-rw	0666	All classes can read/write
-r-xr-xr-x	0555	All classes can read/execute
-r--r--r--	0444	All classes can read

--wx-wx-wx	0333	All classes can write/execute
--w--w--w-	0222	All classes can write
---x--x--x	0111	All classes can execute

All together now

Let's use the examples from the symbolic notation section and show how it'd convert to octal notation

CHMOD commands

Now that we have a better understanding of permissions and what all of these letters and numbers mean, let's take

Permission (symbolic nocation)	CHMOD command	Description
-rwxrwxrwx	chmod 0777 filename; chmod -R 0777 dir	All classes can read/write/execute
-rwxr--r--	chmod 0744 filename; chmod -R 0744 dir	**user** can read/write/execute; all others can read
-rw-r--r--	chmod 0644 filename; chmod -R 0644 dir	**user** class can read/write; all others can read
-rw-rw-rw-	chmod 0666 filename' chmod -R 0666 dir	All classes can read/write

a look at how we can use the chmod command in our terminal to change permissions to anything we'd like! These are just some examples. Using your new-found knowledge, you can set any permissions you'd like! Just be careful and make sure you don't break your system.